

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра автоматизації та управління в технічних системах**

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Програмне забезпечення
інформаційно-комунікаційних систем»
спеціальності 121 «Інженерія програмного забезпечення»
на тему: «Вебзастосунок продажу тематичного медіаконтенту»**

Виконав:

студент IV курсу, групи ІТ-61

Вашеняк Артем Миколайович _____

Керівник:

професор каф. АУТС, д.е.н.,

Шемаєв Володимир Миколайович _____

Рецензент:

зав. каф. Навчально-наукового інституту

інформаційної безпеки

Національної Академії СБ України, к.т.н.,

Бровко Володимир Дмитрович _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2020 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет інформатики та обчислювальної техніки

Кафедра автоматизації та управління в технічних системах

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Програмне забезпечення інформаційно-комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«___» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Вашеняку Артему Миколайовичу

1. Тема проєкту: «Вебзастосунок продажу тематичного медіаконтенту», керівник проєкту Шемаєв Володимир Миколайович, професор кафедри АУТС, затверджені наказом по університету від «07» травня 2020 р. №1081-с

2. Термін подання студентом проєкту 9 червня 2020 року

3. Вихідні дані до проєкту

Мова програмування C#, середовище розробки Microsoft Visual Studio, технологія ASP.NET, система управління базами даних MS SQL Server.

4. Зміст пояснювальної записки

1. Перелік умовних позначень, скорочень та термінів 2. Вступ 3. Огляд та аналіз існуючих рішень 4. Постановка вимог до вебзастосунку 5. Розроблення діаграм 6. Розроблення бази даних 7. Розроблення вебзастосунку 8. Тестування додатку 9. Технічні характеристики 10. Висновки 11. Перелік використаних джерел

5. Перелік графічного матеріалу

UML діаграма класів, діаграма бази даних, UML діаграма послідовності створення товару, UML діаграма використання, лістинг вихідного коду програми

6. Дата видачі завдання «6» березня 2020 року

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1	Аналіз теоретичних матеріалів та вивчення предметної області	29.03.2020	
2	Аналіз існуючих рішень	04.04.2020	
3	Вибір методів та засобів реалізації	16.04.2020	
4	Розробка структури БД та проектування системи	20.04.2020	
5	Виконання графічних документів	13.05.2020	
6	Оформлення пояснювальної записки	13.05.2020	
7	Подання ДП на попередній захист	25.05.2020	
8	Подання ДП рецензенту	4.06.2020	
9	Подання ДП на основний захист	9.06.2020	

Студент

Артем ВАШЕНЯК

Керівник проекту

Володимир ШЕМАЄВ

АНОТАЦІЯ

Вашеняк А.М. Вебзастосунок продажу тематичного медіаконтенту. КПП ім. Ігоря Сікорського, Київ, 2020.

Дипломний проект присвячений розробці системи, що дозволить розповсюджувати аніме продукцію на території України.

Текстова документація містить інформацію про призначення, проектування, архітектуру, функції, розробку, тестування та особливості роботи із програмною системою. Значна увага була приділена вибору технічних засобів для реалізації та проектування, а також засобам реалізації авторизації та аутентифікації, так як система повинна бути безпечною і зручною.

Ключові слова: архітектура програмного забезпечення, C#, ASP.NET, MVC, магазин, аніме, веб-сервер.

Розмір пояснювальної записки – 61 аркуш, що містять 3 ілюстрації та 33 таблиці.

ABSTRACT

Vasheniak Artem. WEB-application for selling themed media content. Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2020.

Diploma project dedicated to develop system that will allows the distribution of anime products in Ukraine.

Text documentation contains purposes, designing, architecture, functions, developing, testing and characteristics of the developed system. Great attention was paid to the choice of means of implementation, design, authentication and authorization.

Keywords: software architecture, C#, ASP.NET, MVC, shop, anime, web-server.

Size explanatory notes - 61 sheet contains 3 illustrations and 33 tables.

Номер рядка	Форм.	Позначення	Найменування	Кіл.	№ екз.	Примі тка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IT61.060БАК.004 ПЗ	Вебзастосунок продажу	61		
6			тематичного медіаконтенту.			
7			Пояснювальна записка			
8	A3	IT61.060БАК.004 Д1	Вебзастосунок продажу	1		
9			тематичного медіаконтенту.			
10			UML діаграма класів			
11	A3	IT61.060БАК.004 Д2	Вебзастосунок продажу	1		
12			тематичного медіаконтенту.			
13			UML діаграма використання			
14	A3	IT61.060БАК.004 ДЗ	Вебзастосунок продажу	1		
15			тематичного медіаконтенту.			
16			UML діаграма послідовності			
17			створення товару			
18	A3	IT61.060БАК.004 Д4	Вебзастосунок продажу	1		
19			тематичного медіаконтенту.			
20			Діаграма бази даних			
21						
22						
23						
24						
25						
Зм.	Арк.	Прізвище	Підпис	Дат	IT61.060БАК.004 ТП	
Розроб.		Вашеняк А.М.			Вебзастосунок продажу тематичного медіаконтенту	
Перевірів.		Шемаєв В.М.				
					Відомість технічного проєкту	
Н. кон.						
Затв.					КПІ ім. Ігоря Сікорського кафедра АУТС група IT-61	
					Лім.	Лист
						1
						63

**Пояснювальна записка
до дипломного проєкту
на тему: «Вебзастосунок продажу
тематичного медіаконтенту»**

Київ – 2020 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ.....	4
ВСТУП.....	5
1 ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	7
1.1 «Pulsar».....	7
1.2 «Fairy Fox»	7
1.3 «Anime Line Group»	8
1.4 Висновки до розділу	9
2 ПОСТАНОВКА ВИМОГ ДО ВЕБЗАСТОСУНКУ.....	11
2.1 Вимоги до функціоналу.....	11
2.2 Вимоги до інтерфейсу.....	30
2.2 Висновки до розділу	31
3 РОЗРОБЛЕННЯ ДІАГРАМ.....	32
3.1 Діаграма класів	32
3.2 Діаграма використання	32
3.3 Діаграма послідовності створення товару	33
4 РОЗРОБЛЕННЯ БАЗИ ДАНИХ	34
4.1 Обґрунтування вибору СУБД	34
4.1.1 MSSQL.....	34
4.1.2 Oracle	35
4.2 Опис структури та полів.....	36
4.3 Висновки до розділу	38
5 РОЗРОБЛЕННЯ ВЕБЗАСТОСУНКУ	39
5.1 Опис середовища та технологій розробки.....	39
5.2 Опис архітектури системи.....	42
5.3 Опис основних функцій серверної сторони	44
5.4 Опис технологій використаних для розробки інтерфейсу системи.....	51

					IT61.060БАК.004 ТП		
Зм.	Арк.	Прізвище	Підпис	Дат			
Розроб.		Вашеняк А.М.			Вебзастосунок продажу тематичного медіаконтенту		
Перевірив.		Шемаєв В.М.					
					Відомість технічного проєкту		
Н. кон.							
Затв.							
					Лім.	Лист	Листів
						2	63
					КПІ ім. Ігоря Сікорського кафедра АУТС група IT-61		

5.5 Створення адаптивності проекту.....	53
5.6 Висновки до розділу	53
6 ТЕСТУВАННЯ ДОДАТКУ	54
6.1 Випробування програмного продукту	54
6.2 Тестування навантаженням.....	56
6.3 Висновки до розділу	57
7 ТЕХНІЧНІ ХАРАКТЕРИСТИКИ.....	58
ВИСНОВКИ.....	59
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60

					ІТ61.060БАК.004 ПЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дат		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

Інтернет-магазин – це електронний ресурс, сайт із певним каталогом товарів та послуг, завдяки якому здійснюється продаж останніх споживачеві. [1]

Інтерфейс користувача – засіб зручної взаємодії користувача з інформаційною системою. Сукупність засобів для обробки та відбиття інформації, якнайбільше пристосованих для зручності користувача. [2]

База даних (БД) – це організована структура, яка призначена для зберігання, зміни та обробки взаємозалежної інформації, переважно великих обсягів. [3]

IDE (інтегроване середовище розробки) – це система програмних засобів, що використовується програмістами для розробки програмного забезпечення. [4]

СУБД (система управління базами даних) – сукупність програмних та лінгвістичних засобів загального або спеціального призначення, що дозволяють створювати, використовувати та керувати базами даних. [5]

ASP.NET – платформа для розробки веб-додатків, до складу яких входять веб-сервіси, програмна інфраструктура, модель програмування. Дана платформа входить в склад .NET Framework. [6]

Windows Presentation Foundation (WPF) – UI фреймворк, що дозволяє створювати десктопний клієнтський застосунок. [7]

Фреймворк - це програмний продукт, що полегшує створення та підтримку технічно складних або навантажених систем. Він, як правило, містить у собі лише базові програмні модулі, а усі інші специфічні для проекту компоненти реалізуються розробником на їх основі. [8]

					IT61.060БАК.004 ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дат		

ВСТУП

Із поширенням інтернет технологій великої популярності набувають інтернет магазини. Вони дозволяють клієнтам здійснювати покупки будь-яких товарів та послуг без безпосереднього знаходження у будівлях цих магазинів, а власникам бізнесів і взагалі відмовитись від великих витрат на додатковий персонал та оренду, що дозволяє обом сторонам заощадити свій час та кошти.

На сьогоднішній день в Україні наявна велика кількість сайтів, що задовольняють потреби майже усіх наших громадян. Проте тематичних інтернет-магазинів, що займалися б розповсюдженням аніме-продукції майже немає. Тому їх створення є актуальним у зв'язку із поширенням даної культури на теренах нашої країни.

Метою дипломного проекту є розробка власного клієнт-серверного програмного продукту для продажу саме такої продукції у мережі Інтернет. На початку роботи було досліджено і порівняно декілька існуючих рішень таких магазинів, що діють на території нашої держави. В результаті дослідження були виявлені наступні риси гарного програмного продукту: простий і зрозумілий у користуванні інтерфейс, детальне розділення продукції за категоріями, поєднання купівлі/доставки товарів із безкоштовним розповсюдженням тематичної продукції, наявність авторизації для різних клієнтів, наявність зручного засобу для управління базами даних.

Призначення дипломного проекту – розробити тематичний інтернет-магазин для продажу аніме продукції.

Основні цілі реалізованого проекту:

- продаж аніме товарів та популяризація такої тематичної продукції на теренах України;
- збільшити та втримати клієнтів завдяки реалізації алгоритмів категоризації та підбору схожої продукції на основі категорій.

					IT61.060БАК.004 ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дат		

Для реалізації цих цілей готовий продукт повинен мати наступний функціонал:

- управління товарами магазину адміністратором (додавання, зміна та видалення)
- можливість реєстрації клієнтів;
- надання різних прав доступу користувачам сайту;
- пошук і перегляд товару та його опису;
- можливість додавати товар до кошику покупок;
- підбір схожих по категорії товарів на основі переглядів та покупок клієнта;
- можливість оцінити товар та залишити відгук про нього.

					IT61.060БАК.004 ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дат		

1 ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

На початку розробки було досліджено декілька існуючих найпопулярніших онлайн-магазинів. У результаті було виявлено сильні та слабкі сторони кожного із них.

На даний момент в Українському сегменті найпопулярнішими є три основних магазини:

- Pulsar[10];
- Fairy Fox[11];
- Anime Line Group[12].

1.1 «Pulsar»

Даний магазин (рисунок 1.1) можна вважати найстарішим (2001 р.) та найбільш відомим в Україні. Його інтерфейс є зручним у користуванні, а кольорова гамма підібрана таким чином, щоб не відштовхувати покупців. Категорії усіх товарів досить вдало розмежовані, тому пошук потрібного не займає багато часу. Також на сайті відсутня реклама інших сайтів, що візуально допомагає втримати нового відвідувача, а кожен продукт містить велику кількість фото із функцією приближення без натиску. Проте при відкритті товару наявний візуально не зручний (хоч і досить детальний) його опис, що погіршує сприйняття покупцем інформації про можливе майбутнє придбання.

1.2 «Fairy Fox»

Інтернет-магазин «Fairy Fox» (рисунок 1.2) є більш молодим (2009 р.) представником таких тематичних площадок розповсюдження продукції. Привабливий на перший погляд інтерфейс приховує у собі не надто вдале

					ІТ61.060БАК.004 ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дат		

розділення товарів на категорії та не часто влучний підбір їх аналогів. Проте самі товари мають зручний опис та вміщують у собі велику кількість своїх фото, що допомагає у візуальному сприйнятті майбутніх покупок.

1.3 «Anime Line Group»

Даний магазин (рисунок 1.3) - також як і попередній - не є першопрохідцем у сфері продажу аніме-продукції, проте він наслідує його недоліки. У ньому розділення товарів виконано не найкращим чином, що негативно впливає автоматичний підбір схожих товарів та ускладнює їх самостійний пошук. Проте на відміну від Fairy Fox, даний магазин при детальному перегляді товару майже не має опису, а також незручний спосіб перегляду фото.

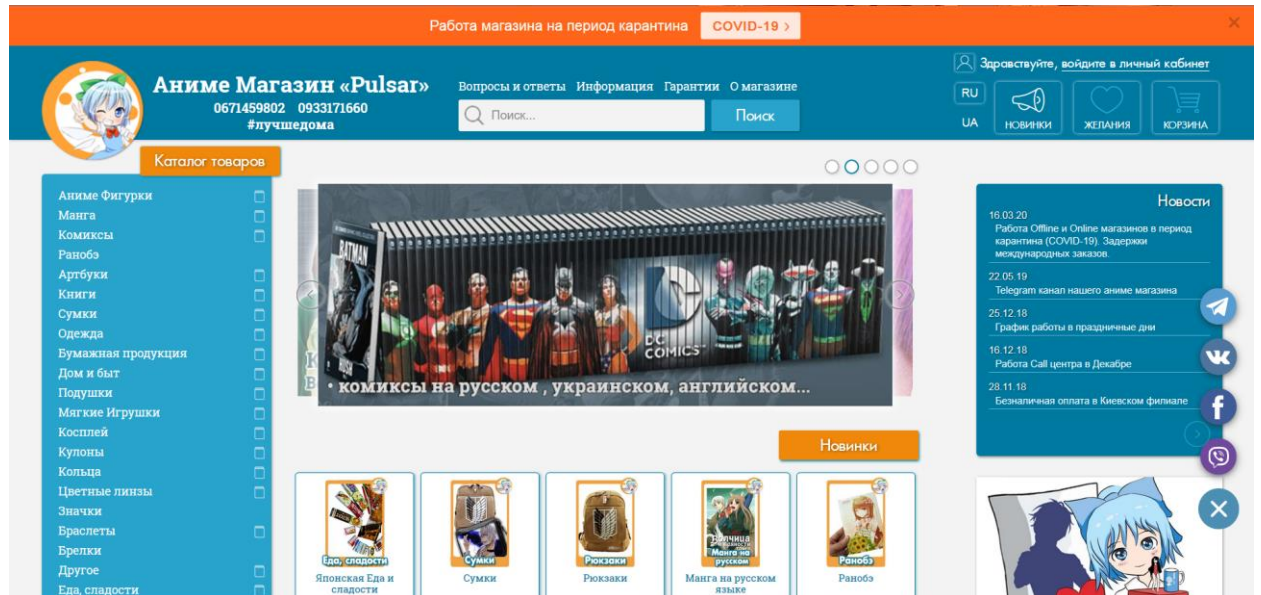


Рисунок 1.1 — Интерфейс магазина Pulsar



Рисунок 1.2 — Интерфейс магазину Fairy Fox

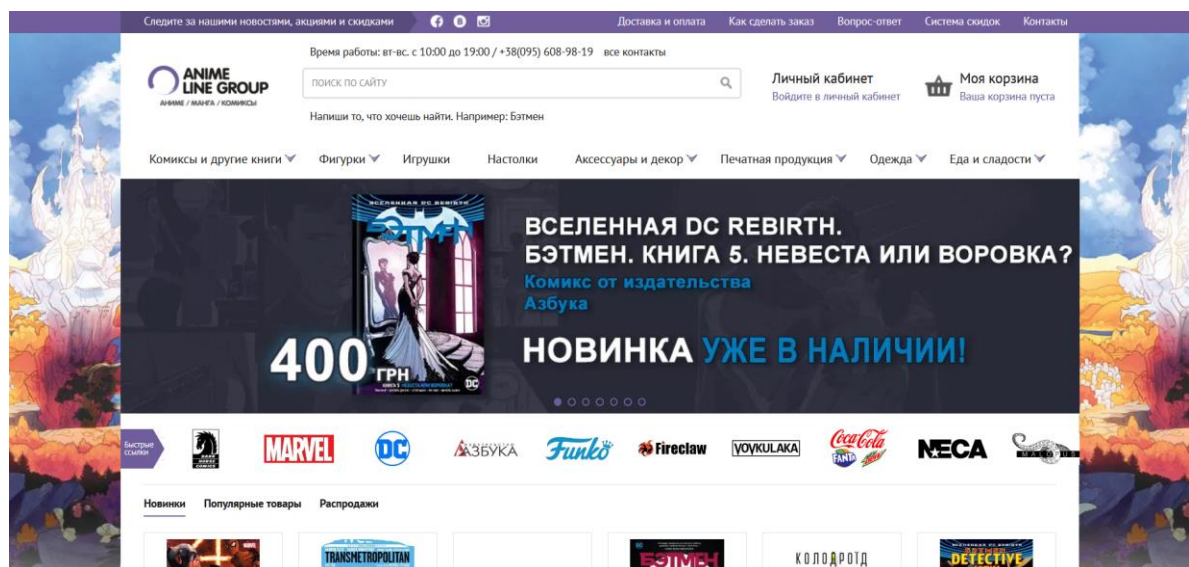


Рисунок 1.3 — Интерфейс магазину Anime Line Group

1.4 Висновки до розділу

У цьому розділі було проведено огляд існуючих на ринку рішень та проаналізовано їх. Нижче, в таблиці 1.1, наведено опис переваг та недоліків розглянутих існуючих рішень.

Таблиця 1.1 – Переваги та недоліки аналогів

Назва	Переваги	Недоліки
Pulsar	<ul style="list-style-type: none"> - привабливий дизайн - вдала категоризація - зручний у пошуку - зручний особистий кабінет 	<ul style="list-style-type: none"> - невелика кількість та не розділення опису товару - не завжди наявне фото
Fairy Fox	<ul style="list-style-type: none"> - привабливий дизайн - містить гарний опис товару та велику кількість якісних його фото 	<ul style="list-style-type: none"> - невдала категоризація - не надто вдалий особистий кабінет
Anime Line Group	<ul style="list-style-type: none"> - місцями привабливий дизайн - зручний особистий кабінет 	<ul style="list-style-type: none"> - невдала категоризація - замалий опис товару - незручний спосіб перегляду фото товару - наявність реклами інших магазинів

2 ПОСТАНОВКА ВИМОГ ДО ВЕБЗАСТОСУНКУ

2.1 Вимоги до функціоналу

Інтернет-магазин для правильного функціонування повинен реалізовувати систему авторизації користувачів для забезпечення необхідного рівня безпеки та збереження інформації користувача при користуванні сайтом. Персональна інформація кожного користувача вказується на етапі реєстрації на сторінці сайту. Завдяки авторизації, користувачам надається можливість створювати замовлення товарів та зберігається уся їхня інформація у конфіденційності від інших користувачів.

Розроблена система розрахована не тільки на звичайних користувачів, що бажають придбати товари, а й на менеджерів даного сайту, котрі будуть розміщувати товари на сайті. Тому внутрішня логіка даної системи була розроблена для користувачів із різними ролями, що надають їм різний функціонал та рівні доступу до даних. Це забезпечує можливість розмежувати функціонал користування звичайних користувачів, менеджерів та адміністраторів.

Більш детальні вимоги по функціоналу наведено в табл. 2.1.

Таблиця 2.1. — Вимоги до функціоналу додатку

Функція	Вимоги
Реєстрація	Потрібно надати користувачу можливість зареєструватися в системі, вказавши свій email, повне ім'я та пароль. Після цього для користувача створюється обліковий запис у системі.
Аутентифікація	Потрібно надати користувачу можливість аутентифікуватись у

Функція	Вимоги
	системі, використовуючи вже створений профіль. Аутентифікація – це процес перевірки відповідності введеного логіну і паролю до логіну і паролю, що збережені у базі даних. Якщо така пара є в БД системи, то користувачу надається доступ до внутрішніх даних та сервісів.
Редагування профілю	Необхідно додати функціонал для редагування персональної інформації користувача. Змінити можна лише певні дані профілю, відповідно наданій користувачу інформації.
Надання користувачу прав менеджера	Потрібно забезпечити можливість адміністратору надати конкретному користувачу права менеджера.
Перегляд сторінки профілю	Необхідно надати можливість переглядати інформацію у своєму профілі
Перегляд списку товарів	Потрібно додати можливість переглянути усі товари, котрі виставлені на огляд на сайті з основною інформацією про них. Дана функція повинна підтримувати розбиття усіх товарів

Змн.	Арк.	№ докум.	Підпис	Дат

IT61.060БАК.004 ПЗ

Арк.
12

Функція	Вимоги
	на різні групи зважаючи на тип та модель кожного із них. Це досягається не тільки завдяки типу та моделі, а й скрипту підбору схожих товарів за назвою.
Додавання нового товару	Потрібно реалізувати можливість створення нового товару із заповненням детальної інформації про нього. Така функція повинна бути тільки на рівні менеджера чи адміністратора.
Перегляд конкретного товару	Необхідно реалізувати можливість перегляду конкретного товару із детальнішою інформацією про нього, де кожен зареєстрований користувач зможе залишити про нього відгук та оцінку, та подивитися відгуки інших користувачів та середню оцінку за продукт.
Редагування інформації існуючих товарів	Необхідно реалізувати можливість змінювати інформацію вже створених товарів. Така функція повинна бути доступна тільки для менеджерів та адміністраторів.

Змн.	Арк.	№ докум.	Підпис	Дат

IT61.060БАК.004 ПЗ

Арк.

13

Функція	Вимоги
Видалення товарів	<p>Необхідно реалізувати можливість видаляти товари з сайту повністю або частково (приховати товар від звичайних користувачів).</p> <p>Така функція повинна бути доступна тільки для менеджерів та адміністраторів.</p>
Оцінка якості товару	<p>Необхідно реалізувати можливість користувачам оцінювати розміщені на сайті товари.</p> <p>На основі їх оцінок повинен формуватися середній рейтинг товару.</p> <p>Дана функція повинна бути доступною тільки для звичайних користувачів.</p>
Залишення відгуку про товар	<p>Необхідно реалізувати можливість користувачам залишення відгуку про товар.</p> <p>Дана функція повинна бути доступною тільки для звичайних користувачів.</p>
Додавання товару у кошик	<p>Необхідно реалізувати можливість додавання нової події з її детальним описом і датою.</p> <p>Дана функція повинна бути доступною тільки для звичайних користувачів.</p>

Функція	Вимоги
Редагування та видалення товарів у кошику	Необхідно реалізувати можливість зміни кількості товарів у кошику та видалення їх із нього. Дана функція повинна бути доступною тільки для звичайних користувачів.
Здійснення покупки	Необхідно реалізувати можливість придбання товарів у кошику «одним чеком» Дана функція повинна бути доступною тільки для звичайних користувачів.
Вихід з акаунта	У користувача має бути можливість виходу з акаунта.
Забезпечення коректності вводу всіх даних	Потрібно перевіряти коректність всіх введених даних, та, при помилках під час введення даних, повідомляти про це користувача.

На основі проаналізованого необхідного функціоналу було створено таблицю сценаріїв використання додатку що допоможуть у реалізації проєкту. Текстовий опис сценаріїв наведено нижче у таблиці 2.2.

Таблиця 2.2 – Сценарії використання системи

Use Case Id	Назва сценарію використання
UC-1	Перегляд сторінки авторизації користувача
UC-2	Авторизація користувача-клієнта
UC-2.1	Перегляд власного профілю клієнтом

					IT61.060БАК.004 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дат		

Use Case Id	Назва сценарію використання
UC-2.2	Перегляд товарів
UC-2.2.1	Додати товар у кошик
UC-2.2.2	Поставити рейтинг та залишити відгук про товар
UC-2.3	Придбати товари у кошику
UC-3	Вихід з системи
UC-4	Створення менеджера
UC-5	Авторизація користувача-менеджера
UC-5.1	Перегляд власного профілю менеджером
UC-5.2	Створення нового товару
UC-5.3	Редагування або видалення товару
UC-6	Авторизація користувача-адміністратора
UC-6.1	Перегляд усіх користувачів
UC-6.1.1	Надання користувачу прав менеджера
UC-6.1.2	Видалення прав менеджера у користувача
UC-7	Перегляд головної сторінки

У таблицях 2.3-2.20 наведено детальний опис кожного з сценаріїв використання системи.

Опис сценарію з ідентифікатором UC-1 представлено у таблиці 2.3.

Таблиця 2.3 – сценарій використання UC-1

Назва	Перегляд сторінки авторизації користувача
Опис	Користувач може переглянути сторінку авторизації
Актори	Неавторизований Користувач
Передумови	
Постумови	Відображено сторінку авторизації

Назва	Перегляд сторінки авторизації користувача
Успішний сценарій	<ol style="list-style-type: none"> 1. Система відображує кнопку «Log in»; 2. Користувач натискає кнопку «Log in»; 3. Система відображує сторінку авторизації
Розширення	

Опис сценарію з ідентифікатором UC-2 представлено у таблиці 2.4.

Таблиця 2.4 – Сценарій використання UC-2

Назва	Авторизація користувача-клієнта
Опис	Користувач може авторизуватися у системі як клієнт на сторінці авторизації
Актори	Користувач
Передумови	Користувач перейшов на сторінку авторизації
Постумови	Користувача авторизовано системою як клієнт
Успішний сценарій	<ol style="list-style-type: none"> 1. Система відображує сторінку авторизації; 2. Користувач обирає «Customer log in»; 3. Користувач натискає кнопку «Register» (якщо немає аккаунта) та заповнює форму після чого натискає «Sign up» або (якщо аккаунт наявний) вводить дані та натискає «Log in»; 4. Система авторизує користувача
Розширення	<ol style="list-style-type: none"> 1. Системою виявлено, що дані користувача введено невірно;

Назва	Авторизація користувача-клієнта
	2. Система відображає спливаюче повідомлення про помилку авторизації; 3. Системою виявлено при реєстрації, що такий аккаунт існує

Опис сценарію з ідентифікатором UC-2.1 представлено у таблиці 2.5.

Таблиця 2.5 – Сценарій використання UC-2.1

Назва	Перегляд власного профілю клієнтом
Опис	Після авторизації користувача як клієнт, система направляє на сторінку профілю, де клієнт має змогу продивитися свій профіль
Актори	Клієнт
Передумови	Користувач має бути авторизований як клієнт
Постумови	Система відображає сторінку профілю клієнта
Успішний сценарій	1. Користувача авторизовано системою як клієнт; 2. Система перенаправляє клієнта на сторінку його профілю
Розширення	1. Система визначила, що користувач не авторизувався у системі як клієнт; 2. На сторінці відображується пропозиція авторизуватися

Опис сценарію з ідентифікатором UC-2.2 представлено у таблиці 2.6.

Таблиця 2.6 – Сценарій використання UC-2.2

Назва	Перегляд товарів
Опис	Клієнт може переглядати усі товари та товари за обраними категоріями/тегами
Актори	Клієнт
Передумови	Користувача авторизовано системою як клієнт
Постумови	Система відображує товари відповідно обраним категоріям/тегам
Успішний сценарій	<ol style="list-style-type: none"> 1. Клієнт обирає вкладку «Products»; 2. Клієнт може обрати усі товари або товари за категорією; 3. Клієнт може налаштувати пошук за тегами; 4. Клієнт може відсортувати товари за назвою або популярністю; 5. Система відображує товари відповідно до заданих критеріїв
Розширення	Система відображує повідомлення, що товарів відповідно до заданих умов не знайдено

Опис сценарію з ідентифікатором UC-2.2.1 представлено у таблиці 2.7.

Таблиця 2.7 – Сценарій використання UC-2.2.1

Назва	Додати товар у кошик
Опис	Клієнт додає обраний товар у кошик
Актори	Клієнт
Передумови	Користувача авторизовано системою як клієнт; Клієнт знаходиться на сторінці товару, що хоче придбати
Постумови	У кошик користувача додано новий товар

Назва	Додати товар у кошик
Успішний сценарій	<ol style="list-style-type: none"> 1. Клієнт знаходиться на сторінці товару; 2. Клієнт натискає кнопку «Add to cart» на сторінці товару; 3. Система демонструє блок вибору кількості товару; 4. Після введення кількості товару клієнт натискає кнопку «Add»; 5. Система додає товар до кошику та підраховує нову загальну вартість товарів у останньому
Розширення	<p>Система відображує повідомлення, кількість товару не введено;</p> <p>Система відображує повідомлення, що даного товару немає у наявності</p>

Опис сценарію з ідентифікатором UC-2.2.2 представлено у таблиці 2.8.

Таблиця 2.8 – Сценарій використання UC-2.2.2

Назва	Поставити рейтинг та залишити відгук про товар
Опис	Клієнт має змогу прокоментувати товар та поставити йому оцінку від 1 до 5
Актори	Клієнт
Передумови	Користувача авторизовано системою як клієнт; Клієнт знаходиться на сторінці товару, який він хоче прокоментувати
Постумови	На сторінці товару з'являється новий відгук та оновлюється середній рейтинг товару
Успішний сценарій	1. Клієнт знаходиться на сторінці товару;

Назва	Поставити рейтинг та залишити відгук про товар
	<p>2. Клієнт має змогу залишити відгук про товар набравши текст у відповідному полі та натиснувши кнопку «Submit»;</p> <p>3. Клієнт має змогу натиснути на одну із 5 зірок для того, щоб залишити свою оцінку товару;</p> <p>4. Сторінка товару оновлена</p>
Розширення	<p>1. Система відображує повідомлення, що коментар пустий;</p> <p>2. Система відображує повідомлення про перебір символів у коментарі</p>

Опис сценарію з ідентифікатором UC-2.3 представлено у таблиці 2.9.

Таблиця 2.9 – Сценарій використання UC-2.3

Назва	Придбати товари у кошику
Опис	Надає змогу користувачеві здійснити покупку товарів, що було додано до кошику
Актори	Клієнт
Передумови	Користувача авторизовано системою як клієнт
Постумови	Користувач придбав товари у кошику
Успішний сценарій	<p>1. Клієнт переходить на сторінку кошику;</p> <p>2. Клієнт натискає кнопку «Make purchase»;</p> <p>3. Виводиться вікно введення даних карти і клієнт після введення даних підтверджує купівлю;</p> <p>4. Клієнт отримує лист, що підтверджує купівлю на пошту;</p> <p>5. Кошик клієнта очищується.</p>

Назва	Придбати товари у кошику
Розширення	<ol style="list-style-type: none"> 1. Система відображує повідомлення, що кошик при купівлі порожній; 2. Система відображує повідомлення, що дані введено не коректно; 3. Система відображує повідомлення, що купівля не була проведена

Опис сценарію з ідентифікатором UC-3 представлено у таблиці 2.10.

Таблиця 2.10 – Сценарій використання UC-3

Назва	Вихід з системи
Опис	Надає змогу користувачеві вийти зі свого аккаунту в системі.
Актори	Клієнт/Менеджер/Адмін
Передумови	Користувач авторизований у системі
Постумови	Користувач виходить із системи та більше не має доступу до контенту, який потребує авторизації
Успішний сценарій	<ol style="list-style-type: none"> 1. Користувач натискає кнопку «Log out» 2. Користувач виходить із системи та більше не має доступу до функціоналу, що потребує авторизації
Розширення	

Опис сценарію з ідентифікатором UC-4 представлено у таблиці 2.11.

Таблиця 2.11 – Сценарій використання UC-4

Назва	Створення менеджера
Опис	Користувач має змогу зареєструвати акаунт у системі із правами менеджера
Актори	Неавторизований користувач
Передумови	
Постумови	Система відправляє запит до адміна на надання користувачу прав менеджера
Успішний сценарій	<ol style="list-style-type: none"> 1. Система відображує сторінку авторизації; 2. Користувач обирає «Manager log in»; 3. Користувач натискає кнопку «Register» та заповнює форму після чого натискає «Sign up»; 4. Система відправляє введені дані на опрацювання адміністратору; 5. Система авторизує користувача як клієнта; 6. Функції менеджера будуть відкриті після схвалення адміністратором
Розширення	<ol style="list-style-type: none"> 1. Система відображує повідомлення, що дані введено не коректно; 2. Система відображує повідомлення, що такий аккаунт існує

Опис сценарію з ідентифікатором UC-5 представлено у таблиці 2.12.

Таблиця 2.12 – Сценарій використання UC-5

Назва	Авторизація користувача-менеджера
Опис	Користувач може авторизуватися у системі як менеджер
Актори	Неавторизований користувач
Передумови	Після реєстрації як менеджера користувач отримав повідомлення на пошту, що підтверджує схвалення його прав менеджера у системі
Постумови	Користувач авторизований системою як менеджер
Успішний сценарій	<ol style="list-style-type: none"> 1. Система відображує сторінку авторизації. 2. Користувач обирає «Manager log in»; 3. Користувач вводить дані та натискає кнопку «Log in». 4. Система авторизує користувача як менеджера.
Розширення	<ol style="list-style-type: none"> 1. Системою виявлено, що дані користувача введено некоректно.

Опис сценарію з ідентифікатором UC-5.1 представлено у таблиці 2.13.

Таблиця 2.13 – Сценарій використання UC-5.1

Назва	Перегляд власного профілю менеджером
Опис	Після авторизації менеджера система перенаправляє його на сторінку власного профілю
Актори	Менеджер
Передумови	Менеджер успішно авторизувався у системі

					IT61.060БАК.004 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дат		

Назва	Перегляд власного профілю менеджером
Постумови	Система відображує власний профіль менеджера
Успішний сценарій	1. Користувач авторизований як менеджер; 2. Користувача перенаправляє на його профіль
Розширення	1. Система відображує повідомлення, що користувач більше не має прав менеджера

Опис сценарію з ідентифікатором UC-5.2 представлено у таблиці 2.14.

Таблиця 2.14 – Сценарій використання UC-5.2

Назва	Створення нового товару
Опис	Користувач створює новий товар у системі
Актори	Менеджер/Адміністратор
Передумови	Користувач авторизований у системі із відповідними правами
Постумови	Створено новий товар
Успішний сценарій	1. Користувач натискає «Manage products»; 2. Користувач обирає «Create new product»; 3. Користувач заповнює дані про товар у відповідних полях нового вікна створення товару; 4. Користувач натискає «Create»; 5. Новий товар створено у системі
Розширення	1. Система відображує повідомлення, що дані про товар введено некоректно; 2. Система відображує повідомлення, що даний товар уже існує

Опис сценарію з ідентифікатором UC-5.3 представлено у таблиці 2.15.

					IT61.060БАК.004 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дат		

Таблиця 2.15 – Сценарій використання UC-5.3

Назва	Редагування або видалення товару
Опис	Користувач має змогу змінити або повністю видалити товар
Актори	Менеджер/Адміністратор
Передумови	Користувач авторизований у системі із відповідними правами;
Постумови	Товар змінено або видалено із системи
Успішний сценарій	<ol style="list-style-type: none"> 1. Користувач натискає «Manage products»; 2. Користувач обирає «Edit products»; 3. Користувач знаходить потрібний йому товар; 4. Користувач натискає «Delete product», після чого знову підтверджує дію після запиту системи, якщо він хоче видалити товар; 5. Користувач обирає «Edit product», змінює дані про товар у відповідних полях нового вікна редагування товару та натискає кнопку «Edit», якщо він хоче змінити товар; 6. Товар змінено або видалено із системи
Розширення	<ol style="list-style-type: none"> 1. Система відображує повідомлення, що дані про товар введено некоректно; <p>Система відображує повідомлення, що даний товар уже існує</p>

Опис сценарію з ідентифікатором UC-6 представлено у таблиці 2.16.

Таблиця 2.16 – Сценарій використання UC-6

Назва	Авторизація користувача-адміністратора
Опис	Неавторизований користувач має змогу авторизуватися у системі, як адміністратор
Актори	Неавторизований Користувач
Передумови	Аккаунт адміністратора уже наявний у системі
Постумови	Користувача авторизовано, як адміністратора
Успішний сценарій	<ol style="list-style-type: none"> 1. Система відображує сторінку авторизації. 2. Користувач обирає «Admin log in»; 3. Користувач вводить дані та натискає кнопку «Log in». 4. Система авторизує користувача як адміністратора.
Розширення	<ol style="list-style-type: none"> 4.1 Системою виявлено, що дані користувача введено невірно. 4.2 Система відображає повідомлення про помилку авторизації.

Опис сценарію з ідентифікатором UC-6.1 представлено у таблиці 2.17.

Таблиця 2.17 – Сценарій використання UC-6.1

Назва	Перегляд усіх користувачів
Опис	Адміністратор має можливість переглядати усіх зареєстрованих у системі користувачів
Актори	Адміністратор
Передумови	Користувач авторизований у системі як адміністратор

Назва	Перегляд усіх користувачів
Постумови	Система відображує список усіх зареєстрованих у системі користувачів
Успішний сценарій	<ol style="list-style-type: none"> 1. Система відображує кнопку «All users»; 2. Адміністратор натискає кнопку «All users»; 3. Система відображує список усіх користувачів системи; 4. Адміністратор має змогу скористатись пошуком конкретного користувача
Розширення	

Опис сценарію з ідентифікатором UC-6.1.1 представлено у таблиці 2.18.

Таблиця 2.18 – Сценарій використання UC-6.1.1

Назва	Надання користувачу прав менеджера
Опис	Адміністратор може надати користувачу, що відправив запит права менеджера
Актори	Адміністратор
Передумови	Адміністратор отримав запит на створення аккаунта менеджера
Постумови	У системі підвищено аккаунт клієнта до аккаунта менеджера
Успішний сценарій	<ol style="list-style-type: none"> 1. Адміністратор натискає кнопку «Requests»; 2. Система відображує усі запити на створення аккаунта менеджера; 3. Адміністратор відкриває запит; 4. Адміністратор натискає кнопку «Confirm upgrade to manager»;

Назва	Надання користувачу прав менеджера
	5. У системі аккаунт підвищено до менеджера
Розширення	

Опис сценарію з ідентифікатором UC-6.1.2 представлено у таблиці 2.19.

Таблиця 2.19 – Сценарій використання UC-6.1.2

Назва	Видалення прав менеджера у користувача
Опис	Адміністратор має можливість видалити права менеджера у користувача
Актори	Адміністратор
Передумови	
Постумови	У користувача відібрані права менеджера
Успішний сценарій	<ol style="list-style-type: none"> 1. Система відображує кнопку «All users»; 2. Адміністратор натискає кнопку «All users»; 3. Адміністратор знаходить менеджера та відкриває його профіль; 4. Адміністратор натискає «Remove rights» та підтверджує це
	5. Система змінює права користувача із менеджера на клієнта
Розширення	

Опис сценарію з ідентифікатором UC-7 представлено у таблиці 2.20.

Таблиця 2.20 – Сценарій використання UC-7

Назва	Перегляд головної сторінки
Опис	Будь-який користувач системи може переглянути головну сторінку
Актори	Будь-який користувач
Передумови	
Постумови	Відображається головна сторінка
Успішний сценарій	1. Користувач потрапляє на сайт. 2. Система відображує головну сторінку.
Розширення	

2.2 Вимоги до інтерфейсу

Зі зростанням доступності та популяризацією технологій починають стрімко зростати і вимоги до кінцевого продукту. Сучасний користувач веб-сайту є дуже вибагливим, тому для втримання та розширення клієнтної бази однією з перших вимог – є дизайн сайту.

Основною вимогою наразі є єдність дизайну на усіх платформах – персональні комп'ютери, телефони, планшети тощо. Кожен користувач, що буде відвідувати сайт повинен мати доступ до єдиного стилю відображення та функціоналу. Це досягається завдяки адаптивній верстці, що у свою чергу втримає майбутніх клієнтів та розширить їхню базу.

Ще одна важлива вимога – це зручність у користуванні вебзастосунком. Сервіс не повинен бути перевантажений функціоналом на одному вікні і усе повинно бути доступно розділено, щоб користувач не загубився та зміг отримати усю необхідну інформацію у найменшу кількість кліків.

Також важливу участь у втриманні користувачів та розповсюдженні вебзастосунку мають кольори. У проєкті застосовано здебільшого нейтральні

кольори – персиковий, чорний та зелений для відображення дизайну усіх форм, а також білим та чорним для тексту.

2.2 Висновки до розділу

У цьому розділі було проаналізовано усі основні вимоги для розробки та візуального оформлення вебзастосунку продажу тематичного медіаконтенту. Було виявлено сильні та слабкі сторони функціоналу. На їх основі було створено макет застосунку що був використаний підчас реалізації проєкту.

					IT61.060БАК.004 ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дат		

3 РОЗРОБЛЕННЯ ДІАГРАМ

3.1 Діаграма класів

Структурна схема зв'язку класів системи зображена на кресленику ІТ61.060БАК.004 Д1. Нижче описано основні класи розробленої системи:

- User – містить основну інформацію про користувача, а саме – його ім'я та прізвище, адресу зареєстрованої електронної пошти та пароль (вони використовуються для входу у систему), а також роль користувача у системі. Також даний клас містить у собі список залишених відгуків та оцінок користувача про товари.
- Product – містить основну інформацію про товар, а саме – його назву, тип та модель (на їх основі ведеться категоризація товару та підбір аналогічних товарів при перегляді), штрих-код та ціна. Також даний клас містить у собі метод повернення середнього рейтингу товару на основі відгуків користувачів.
- Purchase – описує проведення покупки користувачем та містить у собі продукти додані користувачем до замовлення та їх кількість. Також даний клас має у собі метод повернення загальної вартості покупки.
- Rating – містить у собі рейтинг товару, посилання на нього і на користувача, що поставив йому дану оцінку.
- Review – містить у собі відгук про товар, посилання на нього і на користувача, що залишив під ним відгук.

3.2 Діаграма використання

Структурна схема використання різними користувачами системи представлена на кресленику ІТ61.060БАК.004 Д2. На ній показані можливості надані користувачам із різними ролями, а саме для адміністратора, менеджера і звичайного користувача.

					ІТ61.060БАК.004 ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дат		

3.3 Діаграма послідовності створення товару

Діаграму послідовності створення товару можна побачити на кресленику ІТ61.060БАК.004 ДЗ. На ній зображено покроковий процес передачі даних зі сторінки та виклик нової функції для створення товару. На даній діаграмі видно, що процес створення нового товару розпочинається у момент натискання відповідної кнопки менеджером або адміністратором. Після цього зчитується записана у полях даними користувачами інформація та виконується POST запит до серверу із зчитаною інформацією у тілі. Виконаний запит перенаправляється відповідним контролером сервера до сервісу створення товару. Даний сервіс відправляє запит до моделі, звідки він отримує сутність створеного товару, яку він надсилає до бази даних для збереження. Якщо збереження було виконано успішно, то база даних повертає назад збережений у ній товар.

					ІТ61.060БАК.004 ПЗ	Арк.
						33
Змн.	Арк.	№ докум.	Підпис	Дат		

4 РОЗРОБЛЕННЯ БАЗИ ДАНИХ

4.1 Обґрунтування вибору СУБД

В ході аналізу існуючих рішень було розглянуто безкоштовні системи управління базами даних у вигляді цифрової дистрибуції в мережі Інтернет.

Оскільки у наш час існує велика кількість СУБД, при їх аналізі треба звернути увагу на такі критерії:

- зручність інтефейсу;
- гнучкість у системних вимогах;
- збереження інформації;
- простота у користуванні;

4.1.1 MSSQL

MSSQL[15] - це одна із найпопулярніших систем управління базами даних.

Переваги цієї СУБД:

- крос-платформеність;
- простота у встановленні та використанні;
- висока швидкість виконання команд;
- підтримка необмеженої кількості користувачів;
- високий рівень безпеки бази;
- кількість рядків у таблиці досягає 50000.

Натомість програма немає явних недоліків для створення на її базі програми керування таблицями, до того ж вона чудово взаємодіє із C#.

4.1.2 Oracle

Oracle[16] – не набагато менш популярна система управління базами даних.

Її переваги:

- крос-платформеність;
- висока швидкість виконання команд;
- підтримка необмеженої кількості користувачів;
- високий рівень безпеки бази;
- кількість рядків у таблиці досягає 50000;

Натомість, ця СУБД має такі недоліки:

- незручний інтерфейс в інструментах адміністрування;
- система підказок обмежена і громіздка.

Перегляд існуючих популярних магазинів допоміг визначити критерії, котрі необхідні для популяризації власного ресурсу та утримання клієнтської бази від пошуку аналогів у конкурентів. А аналіз вище переглянутих СУБД показав позитивні та негативні якості кожної з них. На основі переглянутих фактів складена наведена нижче таблиця:

Таблиця 4.1 – Порівняльна характеристика СУБД

Критерій оцінки	MSSQL	Oracle
Простота інтерфейсу	+	-
Крос-платформеність	+	+
Висока швидкість виконання команд	+	+

Велика максимальна кількість рядків у таблицях	+	+
Наявні зручні підказки та документація	+	-
Можливість працювати у локальній мережі без доступу до мережі Інтернет	+	+

4.2 Опис структури та полів

На основі проведеного аналізу для проекту мною було обрано систему керування базами даних MSSQL. Структурну схему бази даних можна побачити на кресленику IT61.060БАК.004 Д4.

Нижче в таблиці 4.2 описано усі поля розробленої бази даних.

Таблиця 4.2 — Опис таблиць бази даних

Назва таблиці	Назва поля	Тип даних	Детальна інформація
	id	varchar(100)	Ідентифікаційний номер користувача
	first_name	varchar(100)	Ім'я користувача
	last_name	varchar(100)	Прізвище користувача
	email	varchar(100)	Електронна пошта користувача

Назва таблиці	Назва поля	Тип даних	Детальна інформація
	password	varchar(100)	Пароль користувача
	role	int	Роль користувача
	id	varchar(100)	Ідентифікаційний номер товару
	name	varchar(100)	Назва товару
	type	varchar(100)	Тип товару
	model	varchar(100)	Модель товару
	barcode	varchar(13)	Штрих-код товару
	price	varchar(100)	Ціна за одиницю товару
	order_id	varchar(100)	Ідентифікаційний номер замовлення
	product_id	varchar(100)	Ідентифікаційний номер продукту
	count	int	Кількість купленого товару
	id	varchar(100)	Ідентифікаційний номер замовлення
	user_id	varchar(100)	Ідентифікаційний номер клієнта, що здійснює купівлю

Назва таблиці	Назва поля	Тип даних	Детальна інформація
review	product_id	varchar(100)	Ідентифікаційний номер товару
	user_id	varchar(100)	Ідентифікаційний номер користувача, що залишив відгук
	review	varchar(1000)	Відгук користувача
	product_id	varchar(100)	Ідентифікаційний номер товару
	user_id	varchar(100)	Ідентифікаційний номер користувача, що оцінив заклад
	rating	int	Оцінка користувача

4.3 Висновки до розділу

У цьому розділі було проаналізовано популярні на сьогодні СУБД та, на основі проведеного порівняння було зроблено вибір на користь MSSQL. Також було описано основні поля реалізованої у проєкті бази даних.

5 РОЗРОБЛЕННЯ ВЕБЗАСТОСУНКУ

5.1 Опис середовища та технологій розробки

При розробленні серверної складової вебзастосування, зважаючи на поставленні завдання, за основу було обрано фреймворк ASP.NET MVC Framework та середовище розробки Microsoft Visual Studio 2019. Дане середовище ідеально підходить для створення інтернет-магазину, оскільки у ньому наявна можливість для зручного написання серверної складової на мові С#, написання додаткових скриптів мовою JavaScript та створення сторінок сайту завдяки HTML та CSS.

.NET Framework – це універсальна програмна платформа із відкритим кодом випущена компанією Microsoft, а основою платформи являється загальномовна середовище виконання Common Language Runtime. Саме тому вона підходить для будь-яких мов програмування, проте найкраще себе розкриває у вигляді свого розвитку .NET Core та мові С#. Усі його вихідні файли доступні на GitHub. Даний фреймворк було обрано завдяки тому, що .NET Core забезпечує кроссплатформену розробку і експлуатацію написаних на його основі застосунків. Це дає змогу програмісту відчувати себе більш вільним при написанні коду.

Завдяки тому, що .NET Core – це модульна платформа, усі необхідні для розробки компоненти можуть бути завантажені окремо завдяки NuGet, системі керування пакетами платформ розробки Microsoft. NuGet, а також вбудований в Microsoft Visual Studio – Package Manager Console, забезпечують зручний пошук, встановлення та оновлення пакетів для застосування. Такий підхід полегшує розробку та підтримку кожного окремого проекту.

Центральною складовою проекту є два контейнери – Inversion of Control та Dependency Injection, що, відповідно, надають доступ до інверсії керування та впровадження залежностей. Інверсія керування – це деякий абстрактний принцип, набір рекомендацій для написання слабо зв'язаного коду. Його суть

					IT61.060БАК.004 ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дат		

полягає в тому, що кожен компонент системи повинен бути ізольованим один від одного і не покладатися на деталі конкретної реалізації інших компонентів. Крім даних контейнерів, у проєкті було застосовано декілька модулів ASP.NET MVC:

- AnonymusIdentification – представлений класом System.Web.Security.AnonymousIdentificationModule та відповідає за ідентифікацію запитів навіть коли користувач не авторизований – тобто для анонімних запитів;

- DefaultAuthentication – представлений класом System.Web.Security.DefaultAuthenticationModule та відповідає за встановлення властивості User об'єкта HttpContext;

- FormsAuthentication – представлений класом System.Web.Security.FormsAuthenticationModule та використовується при аутентифікації форм та встановлює значення властивості HttpContext.User, за допомогою якого в застосунку ми можемо отримати аутентифікованого користувача;

- PageInspectorHttpModule – цей модуль підтримує функціональність Page Inspector, яка є в Visual Studio і за допомогою якого виконується відладка HTML и CSS;

- OutputCache – представлений класом System.Web.Caching.OutputCacheModule та відповідає за кешування відповіді користувачу;

- Profile – представляє клас System.Web.Profile.ProfileModule, який зв'язує дані профілю користувача із даними у запиті;

- Session – реалізований класом System.Web.SessionState.SessionStateModule та призначений для зв'язку даних сесії користувача із запитами;

- ScriptModule-4.0 – представлений класом System.Web.Handlers.ScriptModule та призначений для підтримки Ajax—запитів;

- ServiceModel-4.0 – представлений класом System.ServiceModel.Activation.ServiceHttpModule і використовується веб-службами ASP.NET.

Інтегроване середовище розробки Microsoft Visual Studio прекрасно підходить для розробки веб-додатків, тому що підтримує більшість популярних фреймворків та платформ. Розумне автодоповнення коду, інструментарій аналізу коду на помилки, зручна навігація, а також можливість розширеного рефакторингу та форматування коду на основі HTML, CSS та JavaScript значно полегшує та пришвидшує розробку застосування. Також у Visual Studio паявна інтеграція із різними веб-серверами, зокрема з сервером MSSQL, присутній інструментарій для роботи з таблицями бази даних та інструментарій для тестування застосувань.

Бібліотеки для журналювання C# застосування можна створювати на базі NLog. Це дозволяє створювати логи різного рівня:

- Off – ранг, що застосовується тоді коли логування не потрібне тому функція вимкнена;
- Fatal – призначений для виводу усього, до чого можна дотягнутися, так як далі застосунок не буде працювати;
- Error – призначений для звертання уваги розробника. У ньому цікавить лише оточення конкретного місця помилки;
- Warning – відповідає за будь-які неочікувані параметри виклику, дивний формат запиту тобто усе, що може виникнути при позаштатному використанні застосунку;
- Info – разові операції, які повторюються не часто, а саме завантаження конфіга/плагіна, запуск бекапу тощо;

- Debug – призначений для журналювання моментів виклику «великих» операцій, а саме старт/зупинка потоку, запит користувача тощо;
- Trace – виводить усе підряд. На той випадок коли Debug не дозволяє локалізувати помилку. В ньому корисно відмічати виклики різних блокуючих та асинхронних операцій.

Також наявна можливість створення власних рівнів логуювання, що допомагає у більш зручному моніторингу.

Для проєкту також було обрано систему керування версіями Git із його веб-сервісом для розміщення проєктів GitHub.

5.2 Опис архітектури системи

Для максимальної продуктивності застосування та реалізації зручної архітектури було використано RESTful API – стиль архітектури програмного забезпечення та побудови розділених веб-сервісів із масштабуванням. REST – це архітектура, а саме принципи побудови розподілених гіпермедіа посилань. Він надає змогу:

- Масштабувати взаємодію компонентів системи;
- Загальність інтерфейсів;
- Незалежне монтування компонентів;

Проте для того щоб система вважалась побудована по REST архітектурі, необхідно щоб вона відповідала наступним критеріям:

- Client-Server. Система повинна бути розділена на клієнтів та сервери. Розділення інтерфейсів означає, що, наприклад, клієнти ніяк не зв’язані із зберіганням даних, які залишаються всередині кожного сервера, тому мобільність коду клієнтської сторони збільшується. Сервери ніяк не зв’язані із користувачем або станом, тому серверна частина простіша. Сервери та клієнти можуть бути замінені та розробляються, поки не змінюється інтерфейс.

					IT61.060БАК.004 ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дат		

- Stateless. Сервер не повинен зберігати яку-небудь інформацію про клієнтів. В запитах повинна міститися вся необхідна інформація для обробки і, якщо необхідно, ідентифікації клієнта.
- Cache. Кожна відповідь повинна бути відмічена чи є вона кешованою для запобігання повторного використання клієнтами застарілих або некоректних даних.
- Uniform Interface. Єдиний інтерфейс визначає інтерфейс між клієнтами та серверами. Це спрощує та відділяє архітектуру, яка дозволяє кожній частині розвиватися окремо.
- Layered System. Допускається розділити систему на ієрархію шарів, проте з умовою, що кожен компонент може бачити компоненти лише наступного шару.
- Code-On-Demand. Являється опціональною умовою завдяки якій дозволяється завантаження та виконання коду або додатку на стороні клієнта.

Щоб керувати інформацією та передачею було використано протокол HTTPs. Даний протокол прикладного рівня, орієнтовано саме для клієнт-серверної системи. HTTPs підтримує велику кількість різних методів для операцій над ресурсами. Проте у даному проєкті реалізація доступу до бази даних через API в стилі REST було реалізовано чотири основні CRUD-функції через http-методи POST, GET, PUT DELETE.

Метод GET надає змогу отримувати будь-які дані, що були ідентифіковані завдяки URL запиту ресурсу, а також з його допомогою можна розпочати деякий процес. Тому він використовується у запитах на повернення системою даних.

Метод DELETE використовується для надсилання на сервер запиту про видалення. Він використовується системою при надсиланні запитів про видалення товару чи прав користувача.

Метод POST використовується для надсилання на сервер інформації. Наприклад надсилання введених даних при реєстрації або вході у систему, інформації про створений товар чи редагований товар.

Усі дані до сервера надсилаються через HTML-форми у котрі можна заносити інформацію вручну, обирати із списку або натиском на відповідну кнопку. Дані, що внесено у форму, відправляються до серверу на обробку і в залежності від даних, що вніс користувач, вебзастосування може створювати різні веб-сторінки.

5.3 Опис основних функцій серверної сторони

Класи і функції, які відповідають за створення користувача і взаємодію з ним зображені у таблиці 5.1.

Таблиця 5.1 – Функції керування класом «User»

Клас	Функція	Опис
UserController	createCustomer	Приймає запит з клієнту на створення нового користувача і повертає створеного
	updateToManager	Приймає запит з клієнту про зміни ролі «Користувач» на роль «Менеджер»
	getUser	Приймає запит з клієнту на дані про користувача з вказаним ідентифікаційним

Клас	Функція	Опис
		номером і повертає ці дані
	getLoggedInUser	Приймає запит з клієнту на дані про користувача поточної сесії і повертає ці дані
UserController	deleteUser	Приймає запит з клієнту на видалення користувача
UserService	save	Зберігає користувача у системі
UserService	updateToManager	Змінює роль користувача на роль «Менеджер»
UserService	getUser	Повертає користувача по ідентифікаційному номері
UserService	getUserByEmail	Повертає користувача по адресі електронної пошти
UserService	deleteUser	Видаляє користувача із системи

У таблиці 5.2 описано класи і функції, які відповідають за модель «Product» на серверній стороні.

Таблиця 5.2 – Функції керування класом «Product»

Клас	Функція	Опис
	createProduct	Приймає запит з клієнту на створення нового товару з вказаною інформацією і повертає створений товар
	getProduct	Приймає запит з клієнту на дані про товар з вказаним ідентифікаційним номером і повертає ці дані
	getProductList	Приймає запит з клієнту на список товарів з вказаним номером і розміром сторінки і повертає ці дані
	getProductLove	Приймає запит з клієнту на список товарі, що відповідають тематиці покупок клієнта і повертає ці дані

Клас	Функція	Опис
	deleteProduct	Приймає запит з клієнту на видалення товару із системи, проте залишає товарну історію
	save	Зберігає товар у системі
	getProduct	Повертає інформацію про заклад по ідентифікаційному номері
	getProductByName	Повертає інформацію про товар по імені
	getProductsByNameTag	Повертає інформацію про товари по імені та тегу
	getProduct	Повертає список товарів
	deleteProduct	Видаляє товар із системи
	getProductInLove	Список товарів, що найбільше підходять під товар, що переглядають

Класи і функції, що відповідають за модель «Event» на серверній стороні описано у таблиці 5.3.

Таблиця 5.3 – Функції керування класом «Order»

Клас	Функція	Опис
	createOrder	Приймає запит з клієнту на створення нового замовлення із вказаною інформацією та повертає створене замовлення
	getOrder	Приймає запит з клієнту на дані про покупку з вказаним ідентифікаційним номером
	getOrders	Приймає запит з клієнту на дані про покупки, що були здійснені покупцем за весь період користування сервісом
	deleteOrder	Приймає запит з клієнту на видалення замовлення
	saveOrder	Зберігає замовлення покупця у системі

Клас	Функція	Опис
	deleteOrder	Видаляє товар із кошика покупця та повертає змінені дані без одного замовлення

У таблиці 5.4 описано класи і функції, які відповідають за модель «Rating» на серверній стороні.

Таблиця 5.4 – Функції керування класом «Rating»

Клас	Функція	Опис
	createRating	Приймає запит з клієнту на виставлення конкретним користувачем оцінки вказаному товару
	getRating	Приймає запит з клієнту на отримання оцінки конкретного користувача вказаному товару
	saveRating	Зберігає оцінку користувача у системі
	changeRating	Змінює оцінку користувача

Клас	Функція	Опис
	getRating	Повертає оцінку користувача

Класи і функції, що відповідають за модель «Review» на серверній стороні описано у таблиці 5.5.

Таблиця 5.5 – Функції керування класом «Review»

Клас	Функція	Опис
	createReview	Приймає запит з клієнту на додавання конкретним користувачем відгуку вказаному товару
	getReview	Приймає запит з клієнту на отримання відгуку конкретного користувача вказаному товару
	getProductReviews	Приймає запит з клієнту на отримання усіх відгуків вказаного товару
	saveReview	Зберігає відгук користувача
	getReview	Повертає відгук користувача

Клас	Функція	Опис
	getProductReviews	Повертає всі відгуки про вказаний товар

5.4 Опис технологій використаних для розробки інтерфейсу системи

Розмітка веб-сторінок системи виконана стандартизованою мовою розмітки HTML. Вона інтерпретується браузером, та у результаті виводить на екран монітору відповідний текст та/чи медіа.

Документ, що написано мовою розмітки HTML – це лише набір стандартних елементів, тому завдяки цьому створюється лише макет сайту. Для надання сторінці зручного та красивого вигляду для користувача використовується CSS.

CSS – це формальна мова опису зовнішнього вигляду документу, який написано із використанням мови розмітки. За допомогою CSS у проєкт було додано стилізацію усього користувацького інтерфейсу веб-сторінок. Його використання дозволило логічно розділити структуру кожної веб-сторінки, покращило візуальне сприйняття інформації та збільшило зручність користування сторінками. Також, використання CSS для опису зовнішнього вигляду сторінок надає можливість зробити сторінки адаптивними – при відображенні сторінок на різних пристроях із різним розширенням екрану вони будуть виглядати по-різному, що надає змогу користуватися застосуванням не тільки на ПК, а й на мобільних пристроях.

За допомогою скриптової мови програмування JavaScript на сторінки було додано динамічні елементи, що підвищує інтерактивність сторінок.

Також було використано бібліотеку JQuery для збільшення швидкодії та зручності проєкту. Ця бібліотека є орієнтованою саме на взаємодію HTML із мовою JavaScript. Також вона надає легкий доступ до елементів DOM,

об'єктної моделі документу, а також надає зручний у користуванні програмний інтерфейс для роботи з AJAX.

Собою AJAX являє гнучку технологію взаємодії клієнта та сервера. Саме завдяки використанню цієї технології є змога виконувати асинхронні запити до сервера не перезавантажуючи усю сторінку, а оновивши лише деякий елемент. Використання AJAX у купі з ASP.NET MVC виливається у концепцію ненав'язливого JavaScript. Сенс концепції полягає у тому, що весь необхідний код JavaScript використовується не на самій веб-сторінці, а його поміщено в окремі файли *.js. Звідти за допомогою тегу <script> відбувається посилення із веб-сторінки до файлу коду. Це дає змогу відділити візуалізацію від логіки застосунку, а також збільшити швидкодію сайту, так як файл коду кешується системою.

Для виконання AJAX-запитів у ASP.NET MVC наявна така функціональність як AJAX-хелпери:

- Ajax.ActionLink – створює гіперпосилання на дію контролера, при виконанні якого відбувається аjax-запит на цю дію;
- Ajax.RouteLink – схожий на Ajax.ActionLink, проте посилення створюється на певний маршрут, а не на дію контролера;
- Ajax.BeginForm – створює html-форму, яка відправляє аjax-запити для певної дії деякого контролера;
- Ajax.BeginRouteForm – схожий на Ajax.BeginForm, проте аjax-запити відправляються не на дію контролера, а на певний маршрут;
- Ajax.GlobalizationScript – створює посилення на скрипт, що містить у собі певну культурну інформацію;
- Ajax.JavaScriptStringEncode – кодує рядок для використання його у JavaScript.

5.5 Створення адаптивності проекту

У проєкті було застосовано адаптивну верстку, що дозволяє користувачам мати доступ до вебзастосунку із різних платформ отримуючи однаковий функціонал без втрати зрозумілості та єдиним дизайном.

Для відтворення цього було застосовано фреймворк Bootstrap. Даний фреймворк являє собою набір інструментів та засобів налагоджування сайтів та вебзастосунків. Підключення бібліотеки відбувається шляхом збереження файлів із стилями до папки ресурсів проєкту звідки класи Bootstrap мають змогу їх витягнути та використати під час верстки HTML.

5.6 Висновки до розділу

У даному розділі було проаналізовано та описано усі використані під час реалізації технології. Як бачимо, дана система використовує у собі сучасні та популярні технічні рішення. Це надає змогу легко поширювати продукт та отримувати свіжі оновлення для покращення проєкту.

					IT61.060БАК.004 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дат		

6 ТЕСТУВАННЯ ДОДАТКУ

6.1 Випробування програмного продукту

У процесі розроблення проекту було обрано декілька основних функцій, котрі вебзастосунок повинен виконувати, та проведено їх тестування. Нижче у таблицях 6.1-6.4 наведено декілька результатів тестувань.

Таблиця 6.1 – Авторизація користувача

Мета тесту:	Перевірка функції «Вхід у систему»
Початковий стан веб-застосування	Веб-застосування запущене в браузері користувача
Вхідні дані:	Логін та пароль користувача
Схема проведення тесту:	Ввести логін та пароль у відповідні поля, натиснути кнопку «Sign In»
Очікуваний результат:	Відкрита головна сторінка веб-застосування
Стан веб-застосування після проведення випробувань:	Відкрита головна сторінка веб-застосування

Таблиця 6.2 – Додавання товару

Мета тесту:	Перевірка функції «Додавання товару»
Початковий стан веб-застосування	Веб-застосування запущене на пристрої користувача. Користувач авторизований в системі.
Вхідні дані:	Назва товару та його дані.
Схема проведення тесту:	Авторизуватися як менеджер чи адміністратор. Перейти на сторінку усіх

Мета тесту:	Перевірка функції «Додавання товару»
	товарів. Натиснути на клавішу створення товару. Створити товар.
Очікуваний результат:	Відкрита сторінка списку товарів із новим товаром на ній.
Стан веб-застосування після проведення випробувань:	Відкрита сторінка списку товарів із новим товаром на ній.

Таблиця 6.3 – Додавання товару до кошику

Мета тесту:	Перевірка функції «Додавання у кошик»
Початковий стан веб-застосування	Веб-застосування запущене на пристрої користувача. Користувач авторизований в системі.
Вхідні дані:	Усі дані про товар, дані про кошик покупця.
Схема проведення тесту:	Авторизуватися як клієнт. Перейти на сторінку усіх товарів. Натиснути на клавішу додавання товару у кошик. Перевірити наявність товару у кошику.
Очікуваний результат:	У кошику знаходиться доданий клієнтом товар у відповідній кількості.
Стан веб-застосування після проведення випробувань:	У кошику знаходиться доданий клієнтом товар у відповідній кількості.

Таблиця 6.4 – Генерація схожого товару

Мета тесту:	Перевірка функції «Генерування товару»
Початковий стан веб-застосування	Веб-застосування запущене на пристрої користувача. Користувач авторизований в системі.
Вхідні дані:	Будь-який відкритий товар та його дані.
Схема проведення тесту:	Перевірити підбирання схожого товару до того, який відкритий.
Очікуваний результат:	На сторінці відображається схожий не тільки за типом – але і тематикою – товар.
Стан веб-застосування після проведення випробувань:	На сторінці відображається схожий не тільки за типом – але і тематикою – товар.

6.2 Тестування навантаженням

Подібний вебзастосунок зазвичай повинен мати велику кількість користувачів, що призводить до створення великої кількості запитів. Саме тому потрібно провести тестування навантаженням для оцінки потужності системи. Для виконання даного тестування у код проєкту було додано функцію для імітування одночасного входження користувачів у систему.

Дане тестування крайнє необхідне перед початком впровадження проєкту у масове використання, інакше – користувачі зіштовнуться із затримками у відображенні сторінок та виконанні будь-яких дій із ними. Це призведе до того, що розроблене вебзастосування не буде популярним та не буде матеріальної змоги для подальшого його розвитку. Нижче у таблиці 6.5

Таблиця 6.5 – Тестування навантаженням входження користувачів до системи

Назва	Реєстрація нового користувача
Запит на підключення 1 000 користувачів	Усі користувачі швидко змогли увійти до системи без помилок.
Запит на підключення 5 000 користувачів	Усі користувачі змогли увійти у систему, проте із затримкою.
Запит на підключення 15 000 користувачів	Змогли увійти не усі клієнти. Система видає помилку на перевищення часу.

6.3 Висновки до розділу

У даному розділі наведено дані про тестування функціоналу готового проєкту, а також тастування навантаженням. Це дало змогу оцінити стан готовності вебзастосування до впровадження його до публічного використання. Як видно із виконаних тестів – функціонал вебзастосування виконує покладену на нього роль, проте наразі наявна проблема із масовим використанням продукту. Основним рішенням наразі є збільшення потужностей системи та перенесення бази даних на інший сервер.

7 ТЕХНІЧНІ ХАРАКТЕРИСТИКИ

Інтернет-магазин може бути переглянутий на будь-якому комп'ютері та ноутбучі з операційною системою Windows, Linux та Mac OS. Адаптивна версія магазину відкривається на мобільних девайсах платформ IOS та Android. Сайт відкривається та правильно функціонує у всіх сучасних браузерях із підтримкою HTTPS протоколів.

Основні характеристики сайту:

- середній час аутентифікації користувача займає до 5 секунд;
- максимальний час повного завантаження сторінок – 7 секунд;
- середній час роботи алгоритму підбору продукції близько 3 секунд;
- максимальна кількість збереженої інформації в базі даних – 8 ГБ.

					IT61.060БАК.004 ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дат		

ВИСНОВКИ

В результаті виконання дипломного проєкту був розроблений веб-застосунок для продажу тематичного медіа контенту. Готовий проєкт являє собою інтернет-магазин, а саме - сайт. При його створенні було застосовано різні технології для досягнення найкращого результату.

Простий інтерфейс приємно здивує пересічного користувача, а легкість налаштувань - адміністратора.

Архітектурою проєкту було обрано REST веб-сервіс на ASP.NET в основі якої лежить високорівнева мова програмування C#, оскільки така модель є найбільш зручною для реалізації описаної системи. Для забезпечення збереження усіх даних інформаційної системи було використано СУБД MS SQL Server.

Адаптивність вебзастосування на усіх платформах була досягнута завдяки фреймворку Bootstrap. За реалізацію візуальної складової проєкту відповідають HTML та CSS із бібліотекою JQuery. Для динамічної роботи сторінок було застосовано AJAX.

В подальшому я бачу перспективу розвитку такої програми, адже інтернет-магазини починають наздоганяти по популярності звичні нам «магазини-будівлі» (а також завдяки коронавірусу велика кількість людей побачила їх практичність), а аніме індустрія починає непогано розповсюджуватись серед молоді. Отже, можна сказати що даний продукт зацікавить потенційних користувачів і буде комерційно вигідним.

					IT61.060БАК.004 ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дат		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інтернет магазин. URL:
<https://web-lighthouse.com/news/типи-сайтів-частина-3-інтернет-магазин/>
2. Інтерфейс користувача. URL:
https://uk.wikipedia.org/wiki/Інтерфейс_користувача
3. Що таке база даних. URL:
<http://apeps.kpi.ua/shco-take-basa-danykh>
4. IDE. URL:
<https://ru.wikipedia.org/wiki/IDE>
5. Система Управління Базами Даних. URL:
https://ru.wikipedia.org/wiki/Система_управления_базами_данных
6. ASP.NET. URL:
<https://ru.wikipedia.org/wiki/ASP.NET>
7. WPF. URL:
<https://docs.microsoft.com/en-us/visualstudio/designers/getting-started-with-wpf?view=vs-2019>
8. Про фреймворк. URL:
https://web-creator.ru/articles/about_frameworks
9. Шилдт Г. С# 4.0 Полное руководство. [Текст] : Пер. С англ. – М.: ООО «И.Д.Вильямс», 2011. – 1056 с.
10. Pulsar. URL:
<https://www.pulsar.ua/>
11. Fairy Fox. URL:
<https://fairyfox.ua/>
12. Anime Line Group. URL:
<https://anime-market.kiev.ua/>
13. MSDN Microsoft. URL:
<https://msdn.microsoft.com/ru-ru/library/ms233843.aspx>

					IT61.060БАК.004 ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дат		

14. Правові основи організації діяльності інтернет-магазину. URL:
<https://jurist-blog.com.ua/yuridichna-konsultaciya-31-diyalnist-internet-magazinu-poradi-yuristiv-shhodo-kupivli-prodazhu-tovariv-v-merezhi-internet.html>

15. Microsoft SQL Server. URL:
https://ru.wikipedia.org/wiki/Microsoft_SQL_Server

16. Oracle. URL:
<https://ru.wikipedia.org/wiki/Oracle>

17. Архітектура логування. URL:
<https://habr.com/ru/post/135242/>

18. Модулі C#. URL:
<https://metanit.com/sharp/mvc5/19.2.php>

19. REST простими словами. URL:
<https://medium.com/@andr.ivas12/rest-простым-языком-90a0bca0bc78>

20. CSS. URL:
<https://ru.wikipedia.org/wiki/CSS>

21. Введення в AJAX. URL:
<https://metanit.com/sharp/mvc/10.3.php>

					IT61.060БАК.004 ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дат		